Chesapeake:
A 50Gbps Network Processor and Traffic Manager

Brian Alleyne, Sr. Director Product Management

Speeding Toward 100G

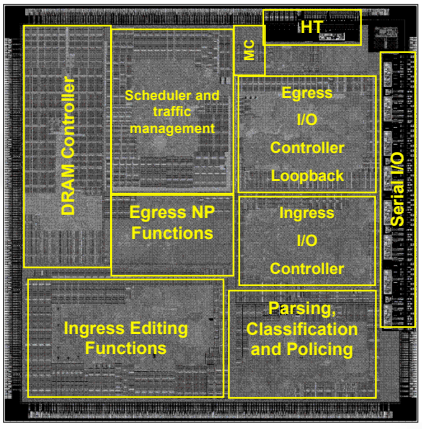
Chesapeake:

A Combined Network Processor and Traffic Manager

Chesapeake Introduction
Architectural Discussion
Pipeline Architecture
DRAM Controller
Open Discussion

"Chesapeake delivers the highest performance of any Network Processor on the market today, while using the lowest power per Gbps in its class"

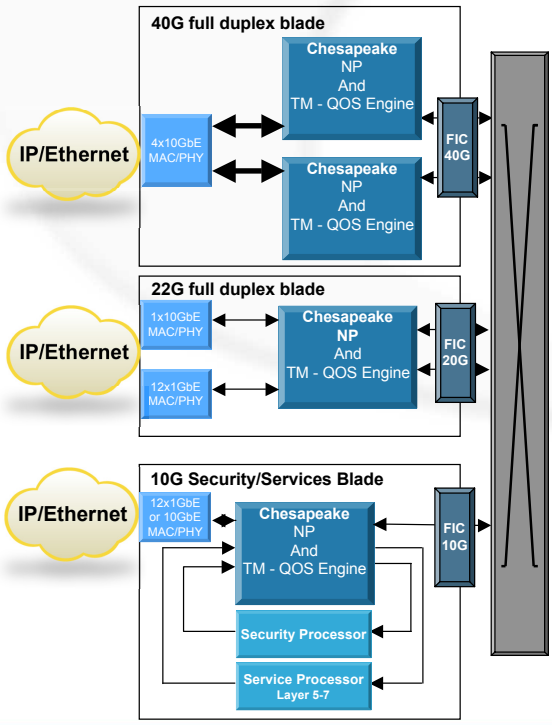
Linley Gwennap, Principal Analyst

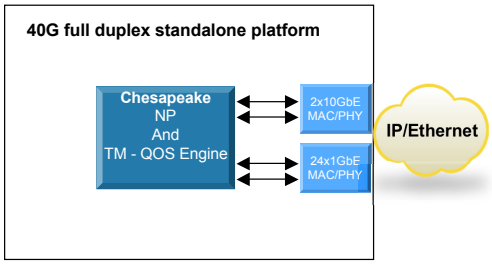


- 50G raw data interfaces
- 125G internal datapath
- 122 million packets per second processing capacity
- TM supports Hierarchical Scheduling and Shaping
- Scalable not dependant only on technology

Speeding Toward 100G

Example Chesapeake Applications

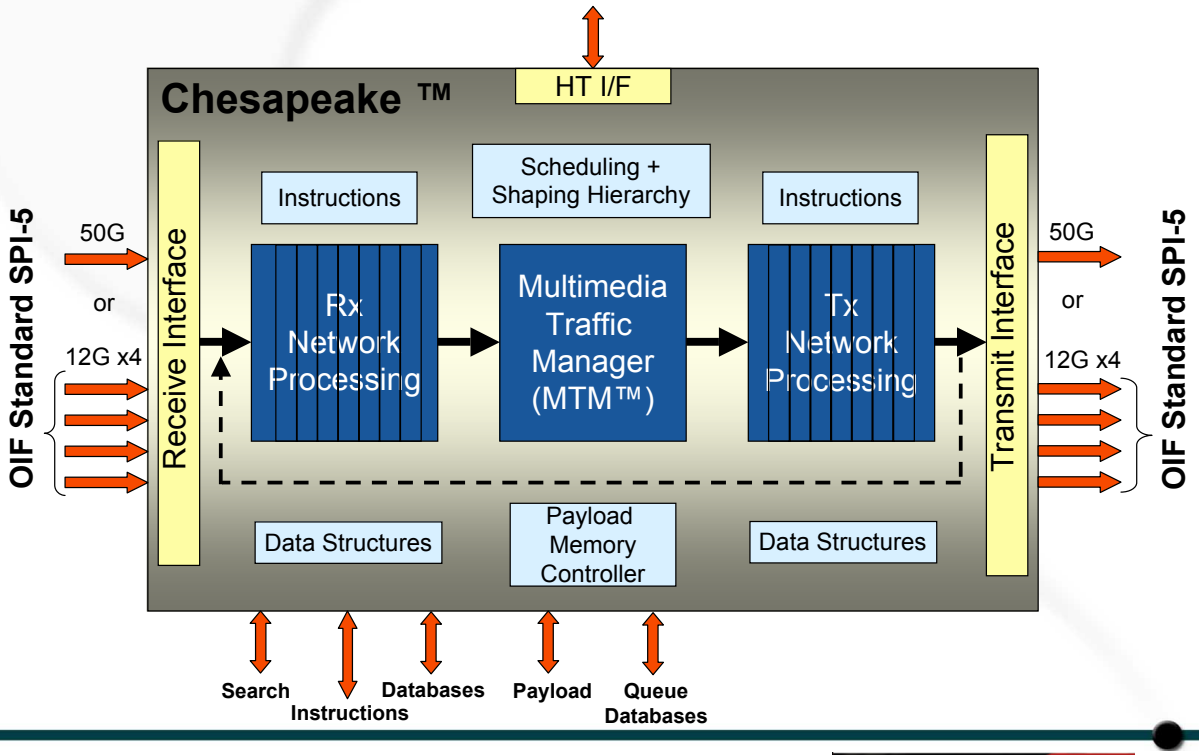




- **Sample Features:**
- **Support for MEF Services: E-Line, E-LAN**
- **Hierarchical Bandwidth Profiles**
- **VLAN Aware L2 Switching, Q-in-Q**
- **PBT**
- **Multicast**
- **VPLS**
- **PWE3**
- **MEF10.1 compliant - Hard-QoS**
- **Multiprotocol Stack**

Speeding Toward 100G

Chesapeake Block Diagram



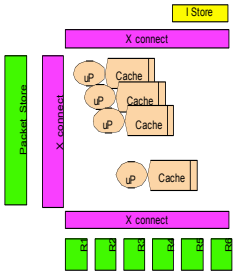
Speeding Toward 100G

The Important Questions

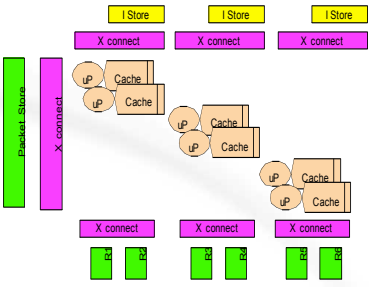
- Why a datapath pipelined approach?
  - Necessary to guarantee deterministic behavior
- What kind of pipeline?
  - Optimize for area
  - Optimize for power
  - Tradeoffs
- How do we sink bandwidth of pipe?
  - Bandwidth is such to require multiple DRAM channels
  - Necessary to guarantee deterministic behavior
- How do all these factors scale?
  - Want to have a family, not a point solution then back to drawing board
  - Don't want to be held hostage only to technology improvements

Network Processor Taxonomy: Parallel Cores

- Sea of Parallel cores
  - General unconstrained programming model
  - Area required for:
    - Resource <-> core interconnect
    - Data and instruction caches
  - Packet arbitration and reordering resources
  - Difficult/Impossible to guarantee deterministic performance

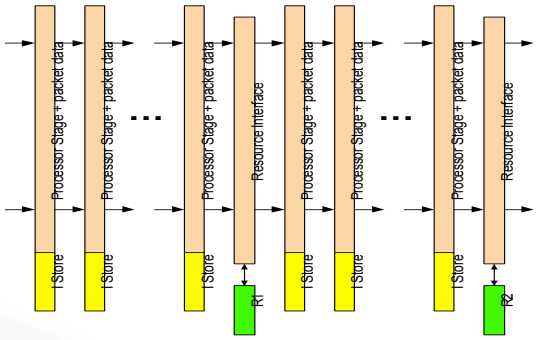


- Functional Partitioning of Parallel cores
  - General unconstrained programming model, but
    - More attention necessary to code partitioning and balancing
    - Packet arbitration and reordering between partitions
  - Area:
    - Reduces but not eliminate resource <-> core interconnect
  - Still difficult/impossible to guarantee determistic performance

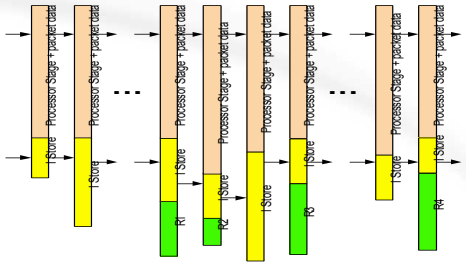


Network Processor Taxonomy: Pipeline

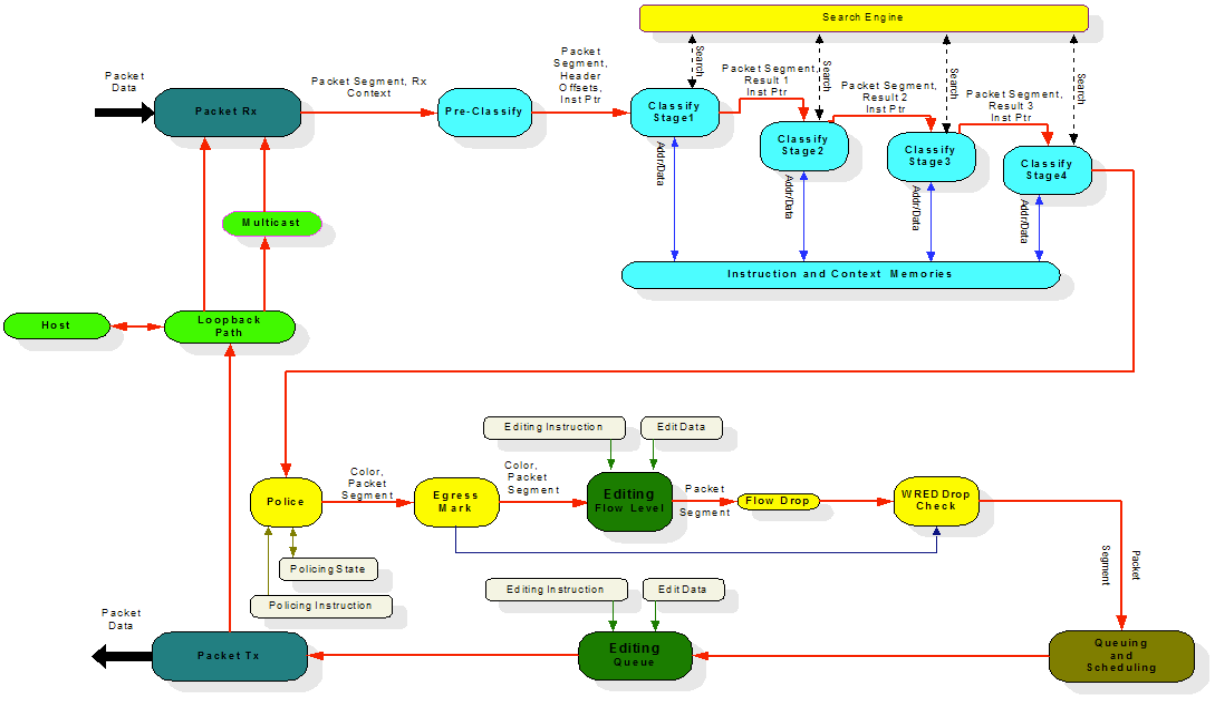
- Replicated Segment Pipeline
  - Programming model
    - No stack based operations
    - No unconstrained loops
    - Harder to share subroutines
    - VLIW tendency in microcode
    - Tight resource planning to pipeline
  - Guaranteed deterministic performance



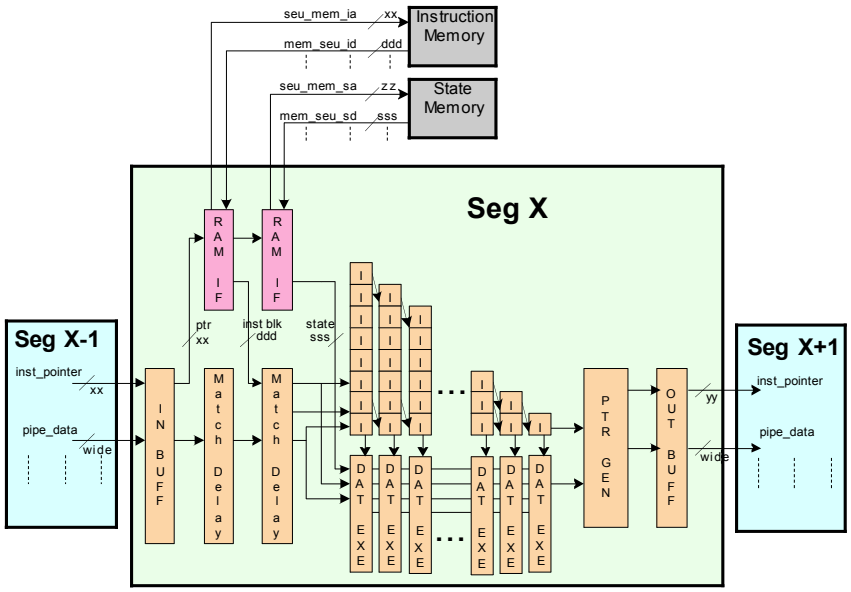
- Functional Specific Pipeline Segments
  - Programming model
    - No stack based operations
    - No unconstrained loops
    - Application specific instruction set
    - More compact code base
  - Guaranteed deterministic performance
  - Area and power optimized



Pipeline Major Segments



Representative Pipeline Segment



Pipeline Considerations

- Chesapeake Baseline
  - Over 300 pipeline stages
  - Pipeline data clocked every 3 clock cycles
  - Core 367Mhz
  - Between 132 and 236 bytes wide datapath
  - 125G operation – 122Mpps pipeline
- Enhancement Areas
  - Architectural
    - Clock pipeline every 2 clock cycles or every 1 clock cycle
    - Extend pipeline length
  - Process: 65nm, 45nm…
    - Shrinking layout gets “free” speed improvement for same design
    - Density increases - more logic for same die area
  - More aggressive design
    - Select Custom Layout
    - Up to 1.5 improvement
    - Note: Have seen pipeline NP design run at 400Mhz core in 180nm process vs. 367Mhz in 110nm at Bay

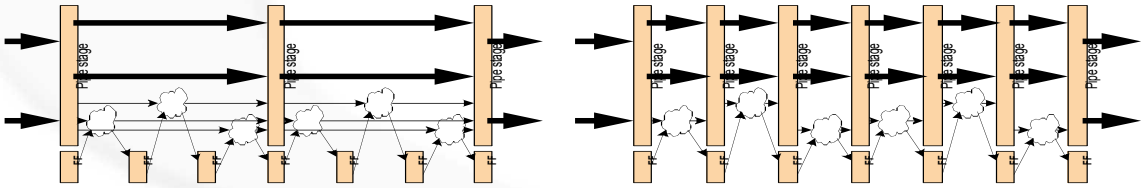
Pipeline performance possibilities

(100G chipset requires: 250Gbps, 244Mpps - only 2x increase)

Architectural

Current:

Possible:



65nm Process (60% smaller)

Enhancement	Pipe		Perf	Relative Area
Arch	3 clk		1.37	.41
Arch	2 clk		**2.06**	.65
Arch	1 clk		**4.11**	.92
Arch + Aggressive	1 clk		**6.17**	.90
Arch + Aggr + Design	1 clk		**9.26**	1.26

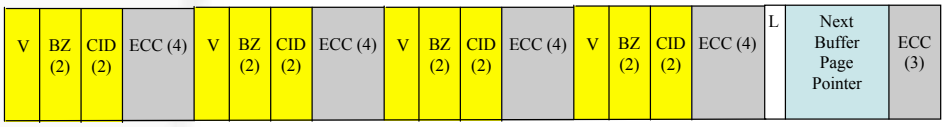
Packet Buffer Challenges

- DRAM Implementation
  - Must stripe across multiple channels to support bandwidth
  - Can trade off utilization for determinism
  - tRC constraints
- Want buffer behavior independent of data behavior
  - Have no control of queue write sequence
  - Cannot rely on statistical behavior not to collide onto a channel
  - Under lighter loads, optimize for behavior
  - Under heavier loads, optimize for performance
  - Cannot stall the pipe before the packet buffer

Packet Buffer

- RLDRAM Bank organization:
  - 4 channels
  - 8 banks per channel
  - 2M buffer page pointers
  - Buffer page pointer controls 4 buffers total (1 buffer per channel)
  - Buffer size is 128 bytes
  - Can write/read 16, 32, 64 or 128 bytes
- Keep track of:
  - Per channel per bank outstanding writes
  - Per channel per bank outstanding reads
  - Free page pointer stack
- Additional per queue link list:
  - Link list of packet lengths

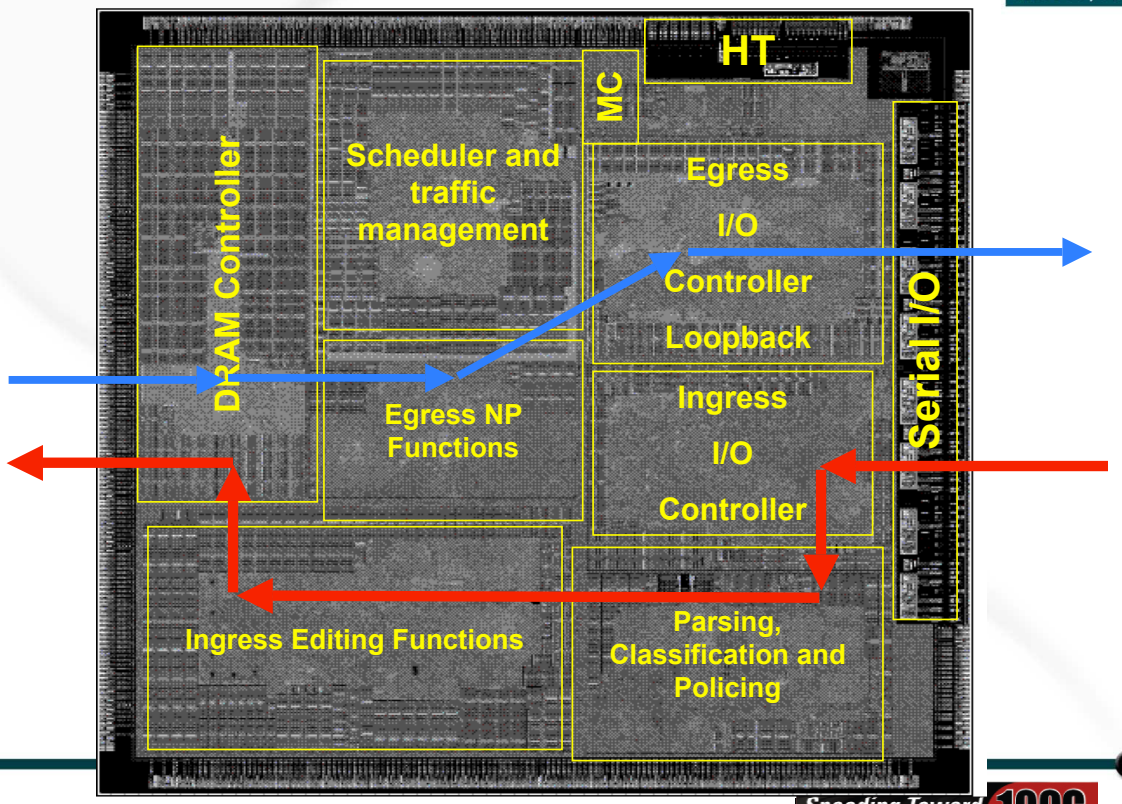
Packet Buffer Algorithm



- V Valid
- BZ Byte Size (128, 64, 32, 16)
- CID Channel ID
- L Last valid row?

- Buffer Page Pointer Data Structure management for writes:
  - If all remaining free channels FIFO full, bump row
  - From available channels, select least loaded channel
  - If all channel FIFOs full, stall pipeline (never happens ☺)
- Channel FIFO algorithm:
  - Usually read requests have priority, but
    - If “high” threshold hit, then write has priority
    - Hysteresis before priority reverts
  - Select oldest eligible bank
    - Potential for out of order writes and reads

No stalling in either direction ☺



Chesapeake Chip Profile

Foundry	Panasonic, Japan
Process	0.11um CMOS
Metals	9 Layers with Cu, FSG
Pads	Signals: 1406 Total Bumps: 5210 IO Bumps: 3197 Core P/G Bumps: 2013
Package	FC-BGA- HiTCE 2401 Balls Size: 50mm x 50mm 25 layers
Voltage	Core: 1.2V HSTL-VDDQ: 1.5V V-RCV: 2.5V
Core Frequency	367Mhz
Performance	125 Gbps 122 Mpps

Die Size (0.11um)	Transistiors: > 270 Million FFs ~ 920 K
Single port Sram	18 Mbits, 89 sets
Dual Port Sram	4.7 Mbits, 90 sets
Power Profile (0.11um)	~16 Watt core ~ 8 Watt terminations
PLL	System PLL: 1, HT PLL: 1, SPI-5 PLL: 8
Test	Full Scan, Boundary Scan, Direct Memory test for repair , Functional Tests Use Full scan with Scan-clk, core-clk to test timing at speed

Summary

- ***=> Deterministic Guaranteed Performance***
- Optimized Pipeline
  - Functionality
  - Performance
  - Power
- Optimized Buffer Management
  - Never stalls pipe
  - Guaranteed for write bandwidth
  - Intelligent on read requests
- ***It scales…***

- L2/L2.5/L3 full solution for Metro Ethernet and Router Applications
- Full featured TM with hierarchical scheduling and shaping

Thank you!

Bay Microsystems, Inc.

Award Winning Global Leader of High Performance Network Processing Solutions

✓ Founding Member Road to 100G Alliance – www.roadto100g.org
✓ Winner 2006 Red Herring Top 100
✓ Fastest Growing Network Processor Supplier
✓ Leading provider of 10Gbps-and-above NPUs

Speeding Toward 100G

Bay Microsystems

- **About:**
  - Founded in 2000 to provide high performance networking solutions
  - Headquarters: Silicon Valley (San Jose, CA)
  - Engineering & Business Development centers: CA, MD, MA

- **Bay Provides:**
  - High Performance Communication Silicon Processors, Systems, & Software
    - Enabling Switching, Routing, ADM, MSPP & Gateway products

- **Our Market Focus and Services:**
  - Network Equipment OEMs & Service Providers
    - Converged Voice, Video, Data & Multimedia for Residential Broadband –"Triple Play"
  - Advanced Enterprise/Government Networks
    - Mission Critical Secure Networking
    - Multiservice Aggregation
    - Grid Computing over the WAN
    - Storage (SAN) extension over the WAN

- **Protected IP Portfolio:**
  - 40+ Filed Patents
  - 19+ Awarded Patents

Speeding Toward 100G